This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A method for defining a bi-cubic spline surface in a computing environment, comprising the steps of:

creating a control mesh with a substantially rectangular structure and allowing T-junctions in at least one parameter direction;

inferring from the control mesh the tensor product B-spline basis functions for each control point; and

computing the surface based on the inferred basis functions and the control mesh.

2. (Original) A method as in claim 1, further comprising the step of determining the basis function for each control point using one non-hierarchical set of rules.

3. (Currently Amended) A method for locally refining a control mesh of a bi-cubic spline surface in a computing environment, comprising the steps of:

~~defining the control mesh having a substantially rectangular structure;~~

inserting a single control point into ~~a pre-image of~~ the control mesh to form a T-junction; and

computing the Cartesian coordinates of the control point and of the neighboring control points using ~~the~~ inferred basis functions such that the bi-cubic spline surface is not geometrically altered.

4. (Original) A method as in claim 3, wherein the step of computing the Cartesian coordinates of the control points further comprises the steps of:

splitting basis functions which have fewer knots than are called for by the control mesh; and

adding control points to the control mesh in locations where basis functions have more knots than are called for by the control mesh.

5. (Original) A method as in claim 3, further comprising the step of creating a sharpness feature in the surface by inserting one or more adjacent partial rows of control points with zero knot intervals thereby creating a sharp crease.

6. (Original) A method as in claim 3, further comprising the step of creating a sharpness feature in the surface by inserting one or more adjacent partial rows of control points with small knot intervals thereby creating a less sharp crease.

7. (Currently Amended) A method as in claim 3 further comprising the step of

providing two surfaces having control meshes that are allowed to contain T-junctions in at least one parameter direction, that are desired to be merged into a single surface;

positioning the two surfaces with common edges that are in close proximity;

performing local refinement on the control meshes of the two surfaces using the T-junctions and adjustment of knot intervals such that a sequence of knot intervals agree along a common boundary edge of the two surfaces; and

joining the control points of the two surfaces along the common boundary.

8. (Cancelled)

9. (Currently Amended) A method for defining bicubic spline surfaces that provides local refinement to control meshes using T-junctions in either or both parameter directions, in a computing environment, comprising the steps of:

specifying knot intervals associated with the ~~spline~~ control mesh;

imposing a local knot coordinate system based on the knot intervals;

inferring local knot vectors for control points in order to produce basis functions for the control points; and

inserting a single control point into the control mesh to form a T-junction without altering the bicubic spline surface.

10. (Original) A method as in claim 9, wherein the step of inserting a single control point further comprises the step of inserting a single control point into the control mesh without altering the bicubic spline surface and permitting partial rows of control points terminating in a T-junction.

11. (Original) A method as in claim 9, wherein the step of imposing a local knot coordinate system further comprises the step of assigning local knot coordinates (s_i,t_i) to the pre-image of each control point P_i.

12. (Currently Amended) A method for subdividing a bi-cubic spline control mesh in order to provide local refinements to control meshes of arbitrary topology in a computing system, comprising the steps of:

imposing local knot coordinate systems on the cubic spline mesh of arbitrary topology;

specifying knot intervals for edges of the cubic spline control mesh;

inserting a T-junction into the cubic spline control mesh;

inferring knot vectors for the T-junction; and

defining basis functions for the T-junction using the knot vectors.

13. (Original) A method as in claim 12, wherein the step of inserting a T-junction further comprises the step of requiring a sum of knot intervals on opposing edges of a face in the cubic spline mesh to be equal.

14. (Original) A method as in claim 12, wherein the step of inserting a T-junction further comprises the step of requiring a T-junction on one edge of a face to be connected to a T-junction on an opposing edge of the face when the sum of the knot intervals on opposing edges of a face in the cubic spline mesh are equal.

15. (Original) A method as in claim 12, further comprising the step of applying shading to the cubic spline mesh that can be viewed by an end user.

16. (Original) A method as in claim 12, further comprising the step of creating a sharpness feature in the spline mesh by inserting a plurality of adjacent rows of control points with zero knot intervals.

17. (Original) A method as in claim 16, further comprising the step of controlling the sharpness feature by placement of the inserted control points and adjusting the knot intervals.

18. (Original) A method for extracting Bézier patches from a T-spline, comprising the steps of:

identifying all Bezier domains by extending all T-junctions two bays;

performing local knot insertion such that each Bezier domain is surrounded by a 4X4 grid of control points;

choosing the knot intervals for the local knot insertion such that pairs of zero knot intervals separate each pair of adjoining Bezier domains.

19. (Currently Amended) A method for merging at least two B-spline surfaces with unaligned knot vectors into a single T-spline, comprising the steps of:

identifying a first B-spline control mesh and second B-spline control mesh;

identifying locations in the first B-spline control mesh for additional control points on an edge configured to align with a knot vector in the second B-spline control mesh which will intersect the edge;

inserting offset control points at each identified location in the first control mesh; and

joining the control points of the first B-spline control mesh with the corresponding knot vectors from the second mesh in order to join the two B-splines and create a T-spline.

20. (Original) A method as in claim 19, further comprising the steps of

identifying locations in the second B-spline control mesh for additional controls points on an edge configured to align with a knot vector in the first B-spline control mesh which will intersect the edge;

inserting offset control points at each identified location in the second B-spline control mesh;

joining the control points of the second B-spline control mesh with the corresponding knot vectors from the first mesh in order to join the two B-splines and create a T-spline.

21. (Original) A method as in claim 19, wherein the step of inserting offset control points at each identified location, further comprises the step of inserting triple knot intervals along the shared boundary in order to provide a C^2 merge.

22. (Currently Amended) A method as in claim 1, wherein the bi-cubic spline surface is a locally refinable tensor product spline surface of any degree in a computing environment. ~~A method for defining a locally refineable tensor product spline surface of any degree in a computing environment, comprising the steps of:~~

~~defining a control mesh with a substantially rectangular structure;~~

~~inferring from the control mesh the tensor product B-spline basis functions for each control point; and~~

~~computing the surface based on the basis functions and the control mesh.~~

23. (New) A method as in claim 1, wherein the step of inferring from the control mesh the tensor product B-spline basis functions for each control point, further comprises the steps of:

imposing a local knot coordinate system, in an area of the control mesh, based on the knot intervals;

generating a ray from the single control point in each of four directions on the control mesh to the two nearest edges of the control mesh in order to infer local knot vectors for control points; and

inferring basis functions for the control points using the knot vectors.